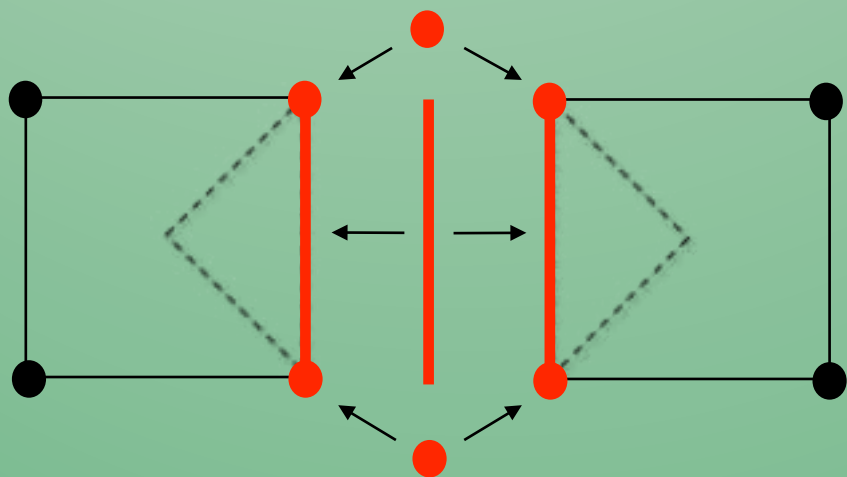
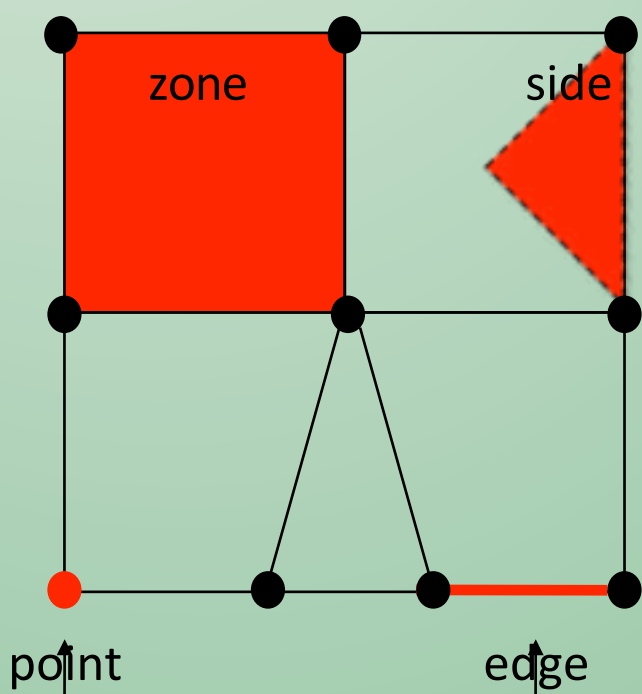
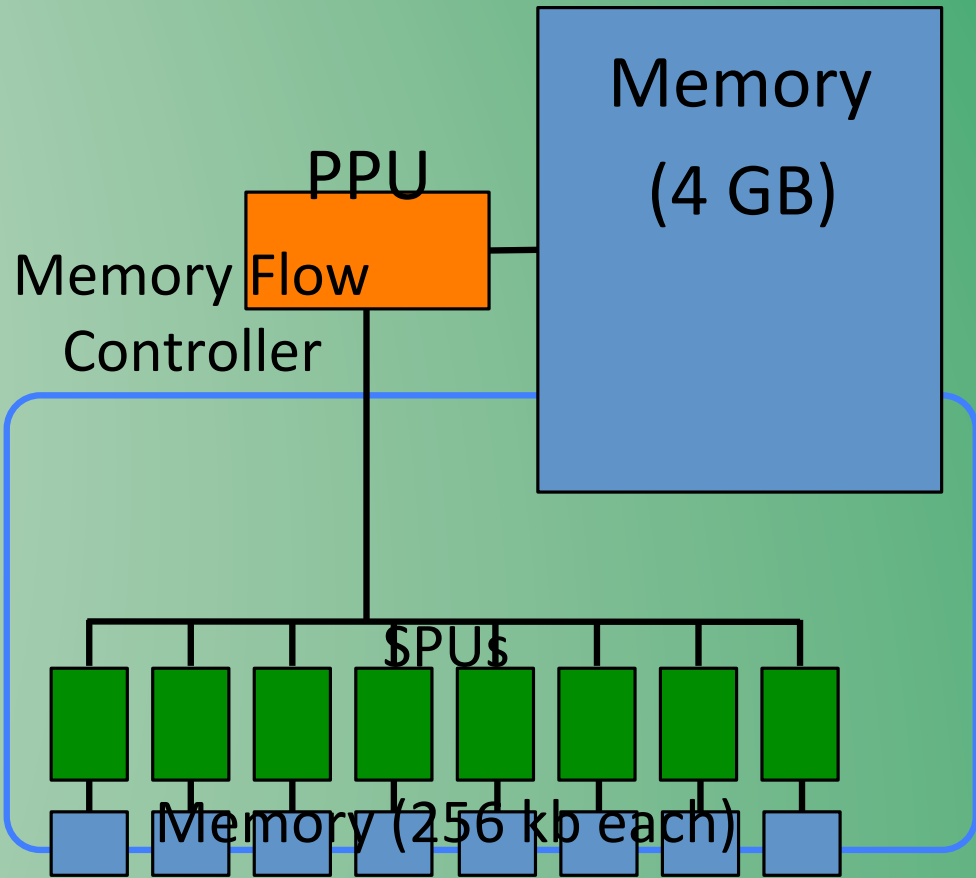


Unstructured Mesh Physics on the Cell Broadband Engine

Unstructured mesh algorithms are particularly challenging to implement on the CBE. For optimal CBE performance, a problem must be divided into many small, independent chunks that can fit into SPU local memory. But this is difficult to do for unstructured meshes.



This approach was used to implement a FLAG compute kernel on the SPU. Three standard test problems were run (see results at right). The accelerated kernel showed **an average speedup of over 5x** compared to the original Opteron kernel.



For example, in the FLAG rad-hydro code, most computation is done on sides (see diagram at left).

Connectivity arrays are used to get data from zones/points/edges as needed.

Because the mesh has no structure, chunks of sides generally do not correspond to chunks of points or edges.

To address this problem, we require the PPU to replicate, or **scatter**, point and edge data onto each side. Then chunks of sides can be sent to the SPU for compute, along with the replicated point and edge data. When compute is done, the PPU will **gather** the replicated data back to points and edges.



A similar approach may also be useful on other emerging architectures with multiple threads and reduced memory on core, such as GPU and many-core.



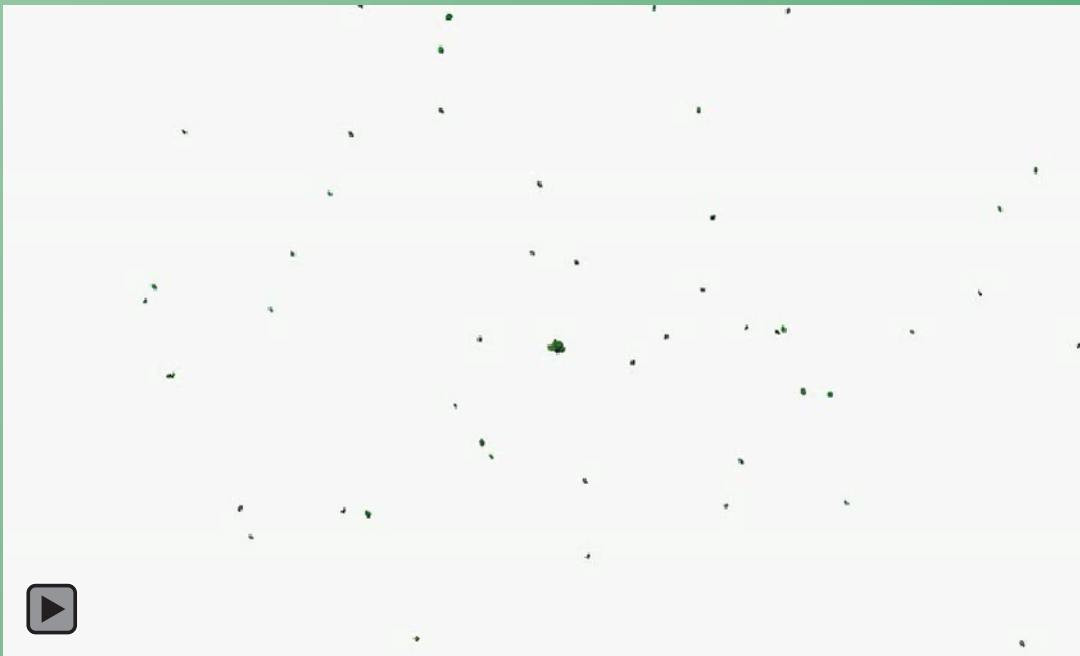
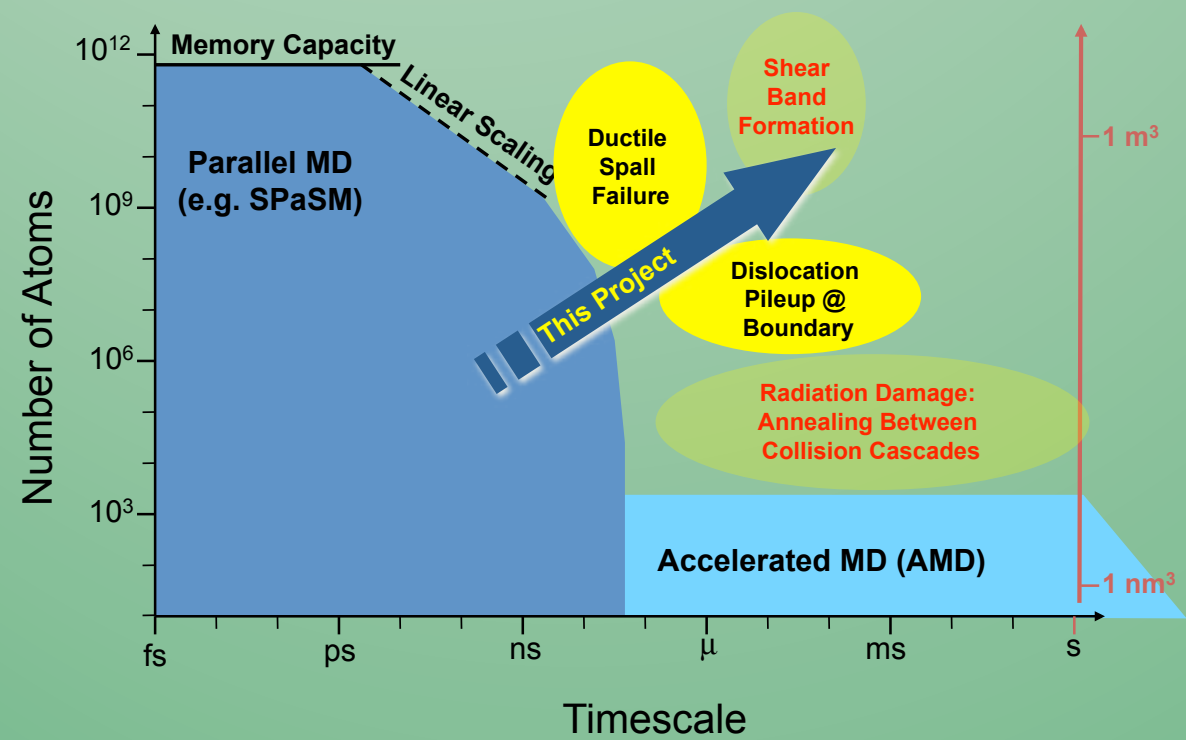
Spatio-Temporal Frontiers of Atomistic Simulations in the Petaflop Computational World

Goal: Develop a transformational atomistic simulation capability to enable studies of previously inaccessible materials science issues, by bringing together world-class LANL leaders in both large-scale and accelerated molecular dynamics algorithms.

Initial target applications on which we will demonstrate this capability include:

- Dislocation pileup against a grain boundary: determine the nature of the critical event controlling material strength, a fundamental long-standing problem; and
- Spall failure: develop an improved understanding of void nucleation, growth, and coalescence dynamics at length and time scales that cannot be directly probed experimentally, enabling development of a science-based model.

Both SPaSM (Scalable Parallel Short-range Molecular dynamics) and the ParRep AMD (Accelerated Molecular Dynamics) codes were among the initial set of Roadrunner Open Science application codes, and demonstrated excellent hybrid performance and scalability.



Algorithm Developments

Approach #1: Local bond-boost hyperdynamics

- Massively parallel implementation



Approach #2: Concurrent MD-AMD via embedded AMD regions

- The embedding approach has been used to study void growth and linkup

Materials science challenges

- Dislocation-grain boundary interactions:
 - Continue efforts to accelerate transmission event in a reduced-size (quasi-2D) model.
 - Identify transmission mechanisms and kinetics, and their dependence on applied stress (strain rate) and temperature.
- Ductile spall failure:
 - Extend studies of different defects (e.g. initial void sizes, grain boundaries, or other heterogeneous nucleation sites) down to experimentally relevant strain rates of 10^4 - 10^5 s⁻¹.
 - Use AMD+embedding scheme to study void growth dynamics (dislocation propagation velocity), coalescence of multiple voids, and interactions of emitted dislocations with other nearby heterogeneities (e.g. a grain boundary).

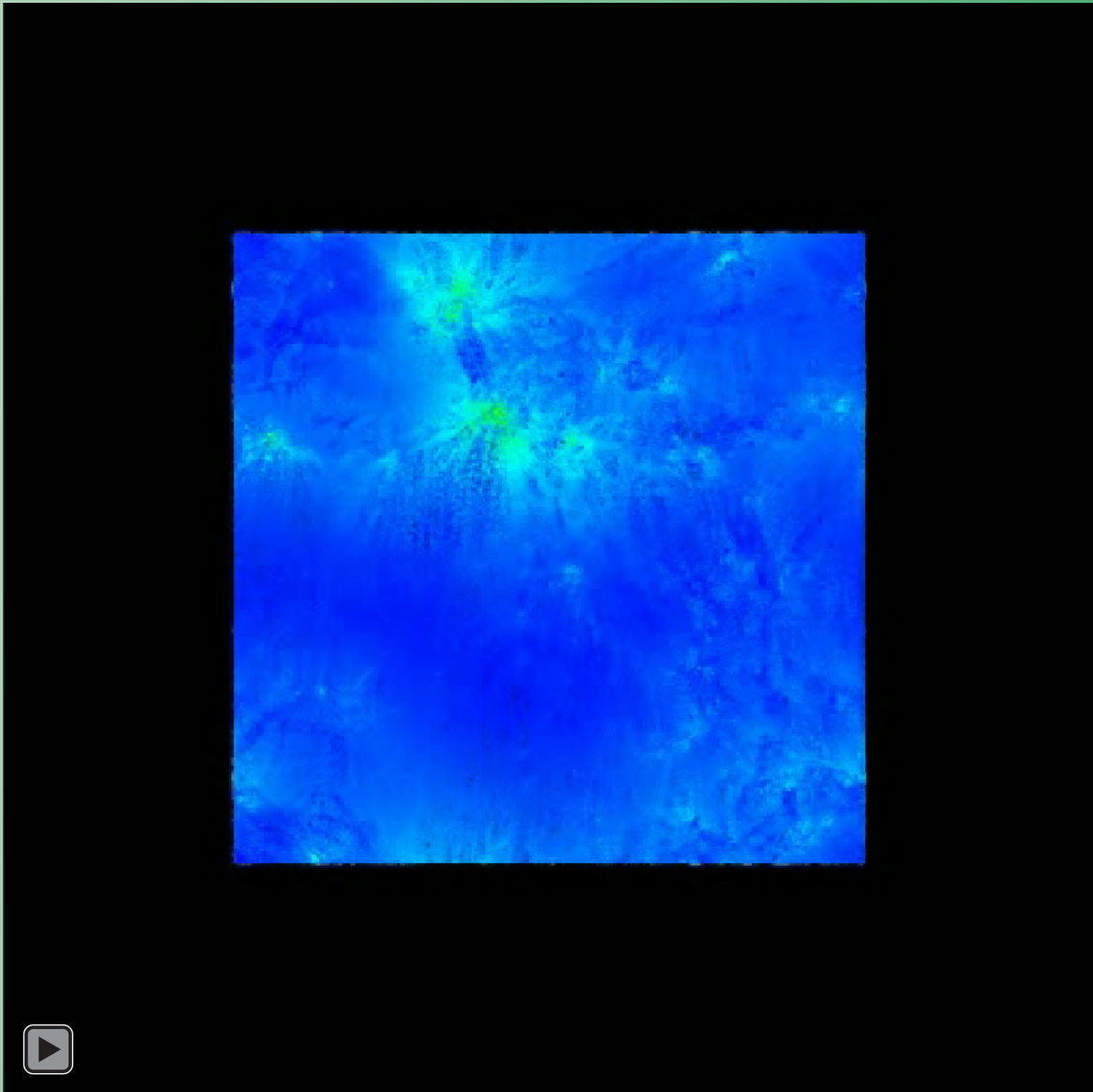


PI: Tim Germann (T-1)
 Co-PIs: Art Voter (T-1), Jim Hammerberg (XCP-5)
 T-1: Shao-Ping Chen, Shiyu Du (PD) , Brad Holian, Danny Perez
 MST-8: Dick Hoagland, Ben Liu, Steve Valone, Jian Wang
 XCP-5: Ramon Ravelo (UTEP LTVSM), Davis Tonks
 CCS-2: Steve Sintay (CMU GRA), Sriram Swaminarayan
 P-24: Sheng Luo, Bedri Arman (TAMU GRA)

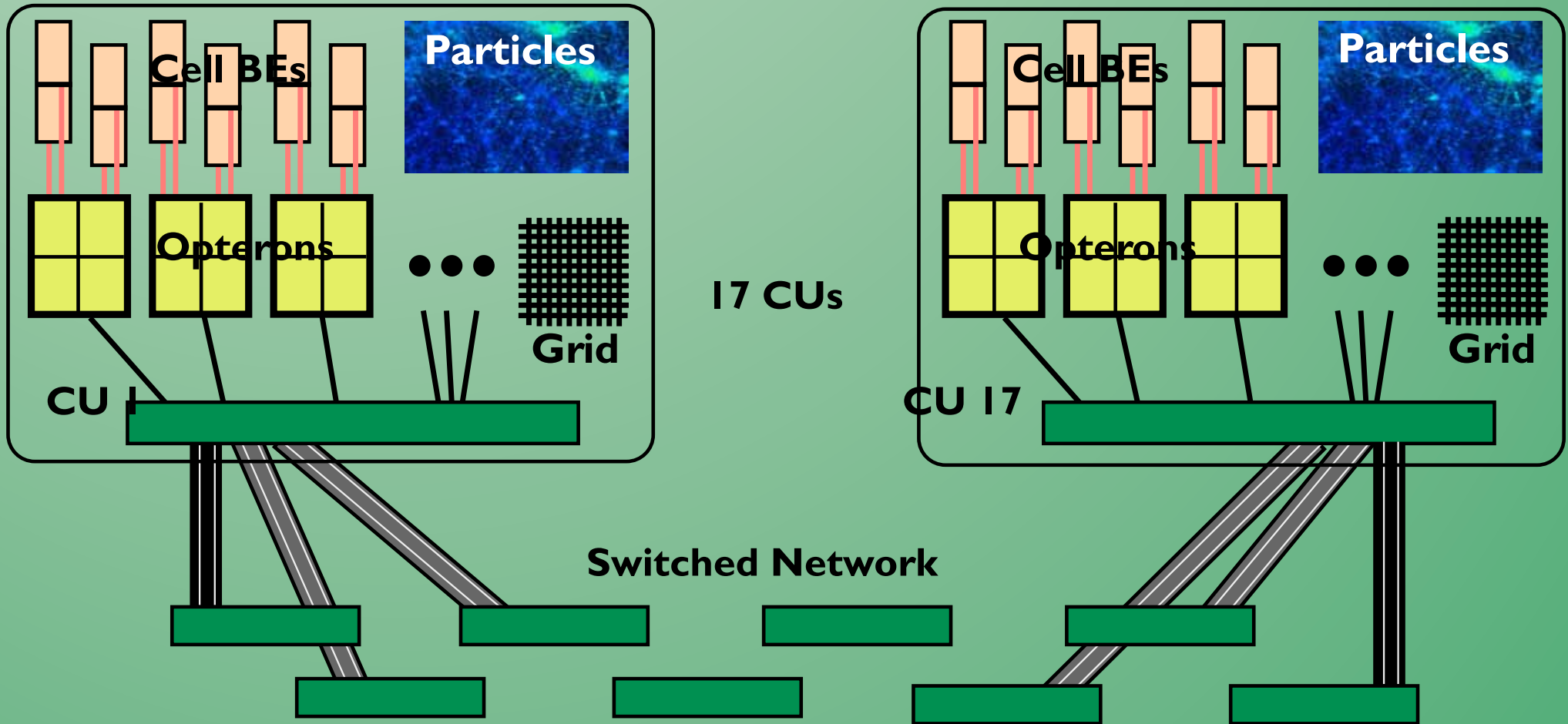


HACC: Hardware Accelerated Cosmology Codes

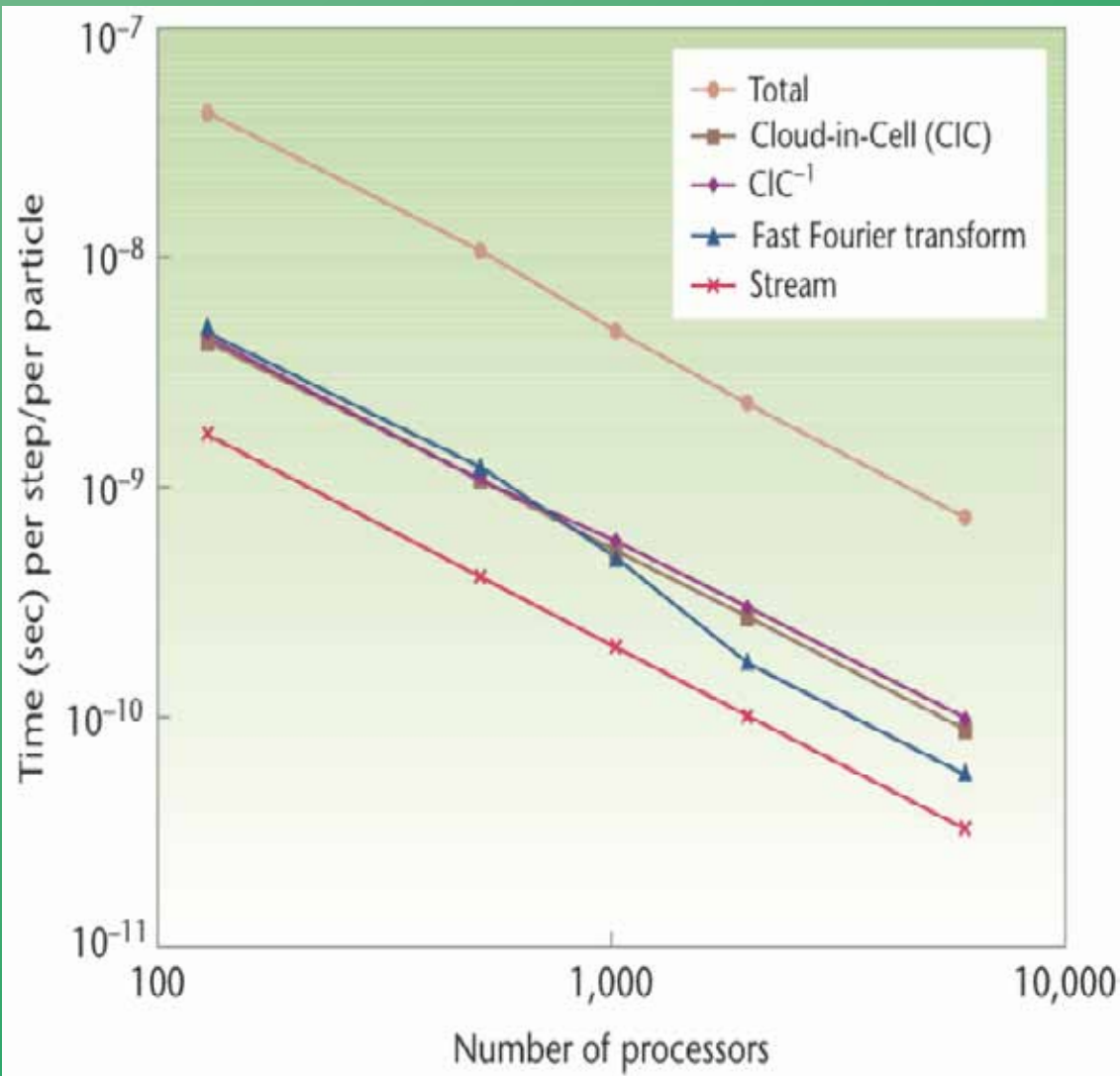
Recent remarkable progress in cosmology is driven by large-scale sky survey observations. The size and complexity of new datasets poses a major challenge; to address it, we have developed the HACC framework (Hardware Accelerated Cosmology Codes) for heterogeneous systems, portable across different architectures. HACC runs on LANL’s Cell-accelerated Roadrunner, and has been ported to GPU-accelerated systems. A hybrid particle/grid algorithm overcomes communication and performance bottlenecks by combining algorithmic features with overloaded data layouts.



Structure Formation in the Universe



- HACC uses grids for long-range calculations and particles for short-range calculations
- On Roadrunner, the CPU and Cell layers are memory-balanced but the Cell layer dominates computational performance, hence particles live at the Cell layer
- In GPU clusters, node memory is dominated by CPUs and all information resides there; particle interactions are streamed through the GPUs
- Inter-node communication is minimized by particle overloading (a particle ‘ghost zone’) with intermittent refreshes using nearest neighbor communication



Weak scaling results for HACC on Roadrunner: the code scales across the entire machine



Katrin Heitman, heitman@lanl.gov, 505-665-9035
Salman Habib, habib@lanl.gov, 505-667-5265

ForOpenCL: Tools for Parallel Acceleration of Fortran Applications with OpenCL

ForOpenCL provides programmers with the tools necessary to parallelize Fortran applications for GPU and multi-core acceleration. It includes a set of bindings for building and running OpenCL kernelsfrom Fortran. It also provides the capability to transform some Fortran procedures automatically with source-to-source transformations using ROSE. For example, pure-elemental Fortran functions can be transformed directly to OpenCL kernels by ForOpenCL.

Step 1: Create Fortran Function	<pre> ! add two scalars ! elemental function add(a,b) real :: a, b, add add = a + b end function </pre>
Step 2: Automatically Transform to OpenCL kernel using ForOpenCL.	<pre> /* transformed OpenCL kernel */ __kernel void add(__global float * A, __global float * B, __global float * C) { const uint x = get_global_id(0); const uint y = get_global_id(1); const uint k = x + y*get_global_size(0); C[k] = A[k] + B[k]; } ! declare arrays and CL Objects real, target, dimension(NX,NY) :: A, B, C type(CLBuffer) :: d_A, d_B, d_C ... ! create OpenCL device (GPU or CPU) status = device%init(GPU) ! create memory buffers d_A = device%createBuffer(4*NX*NY,c_loc(A)) ... ! add arguments to the kernel status = kernel%setKernelArgMem(0, d_A) ... ! call the kernel status = kernel%run(16,16,NX,NY) ! copy results from the device h_C = d_C%map(CL_MAP_READ) call c_f_pointer(h_c, p_C, shape(C)) print *, p_C(1,:) </pre>
Step 3: Modify application to call the OpenCL kernel.	
Standard Fortran transformational functions like CSHIFT can be used with ForOpenCL for more complicated algorithms like multipoint stencil operations.	<pre> ! compute the 3-point, centered average ! of array B along the first dimension ! A = B + CSHIFT(B, SHIFT=-1, DIM=1) & + CSHIFT(B, SHIFT=+1, DIM=1) A = A/3.0 </pre>

Special versions of the Fortran transformational functions can be used (not yet available in ForOpenCL) to automatically parallelize over MPI nodes. These functions will allow full parallelization across the architectural hierarchy (GPUs and MPI nodes) with data-parallel functions like CSHIFT. ForOpenCL requires a Fortran 2003 compiler with at least objects and C interoperability. transformations using ROSE. For example, pure-elemental Fortran functions can be transformed directly to OpenCL kernels by ForOpenCL.



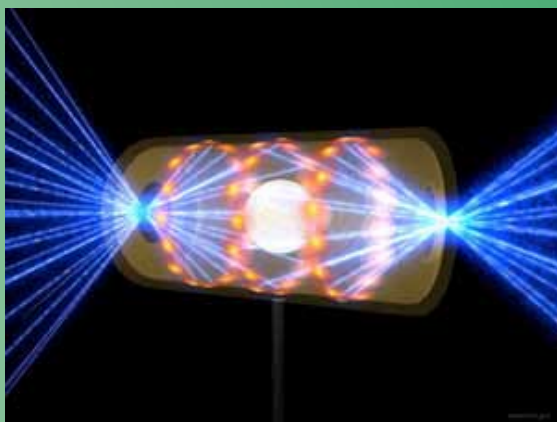
Understanding Stimulated Raman Scattering (SRS) Effects for Nuclear Fusion Research

In January 2011, laser fusion ignition experiments begin at the National Ignition Facility (NIF)

- Aim 192 laser beams into a gold cylinder called a hohlraum
- 1.8 Million Joules of energy are deposited in 10 ns into the hohlraum - the walls are heated up to several million degrees
- The hot walls radiate X-rays, which are absorbed by and compress the capsule—the fuel density becomes 100x the density of lead and temperatures of 100 million degrees
- Deuterium-tritium fusion ignites

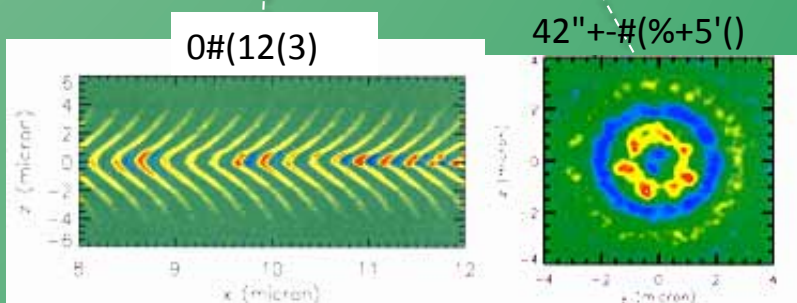
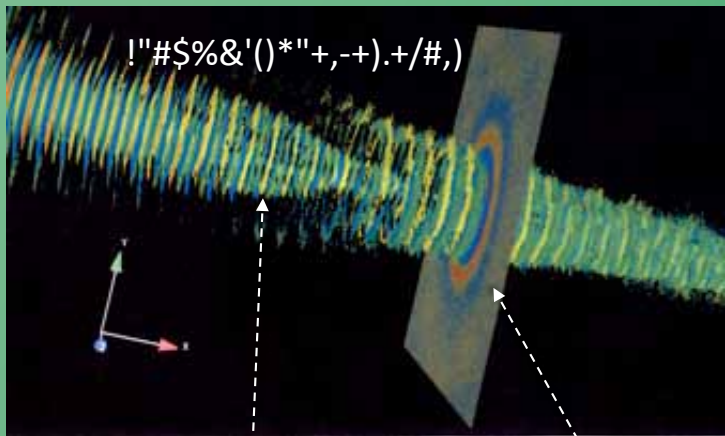
However , getting all the laser energy in is tricky

- Lasers shining through ionized gas (plasma) can experience stimulated Raman scattering (SRS)
- If SRS is too great, ignition fails:
 - Energy is lost
 - Compression is not symmetric
 - Hot electrons are made that pre-heat the core and make it hard to compress
- SRS poses one of the biggest uncertainties in laser fusion

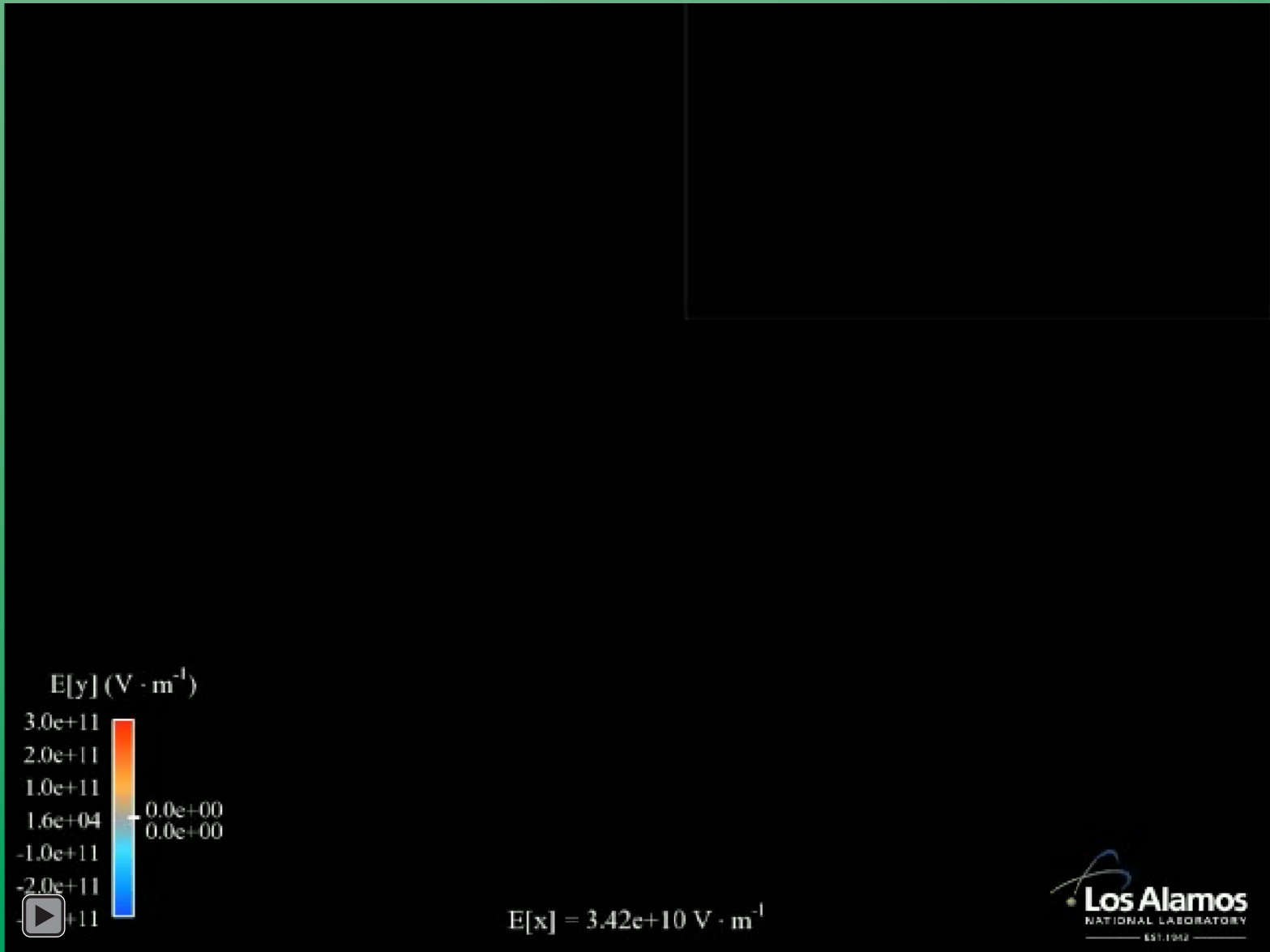


Electron trapping physics explains the curious properties of SRS

- Electron trapping reduces wave (Landau) damping
 - SRS grows faster
 - Onset threshold is lower
- Trapping takes energy from the wave, which lowers the wave frequency and phase speed
 - Wave bending
 - Breaks into filaments
- These processes destroy spatial coherence, cause SRS to saturate, and limit how much SRS backscatter can occur.



Roadrunner allows us to reduce uncertainty in threshold intensity in NIF beams

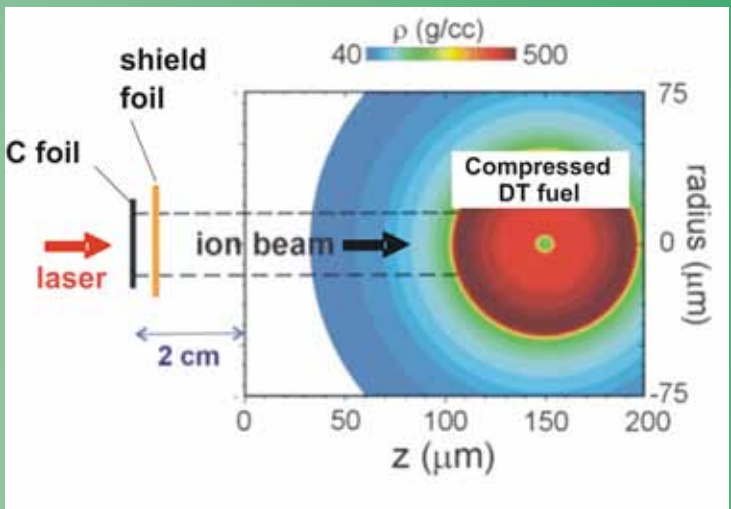


SRS saturation in 3D VPIC simulations under conditions relevant to NIF

Lin Yin, lyin@lanl.gov

Ion Acceleration for Ion-Driven Fast Ignition

Fusion experiments at the National Ignition Facility (NIF) use lasers to generate X-rays that compress a Deuterium-tritium capsule. However, due to instabilities caused by laser-plasma interactions, the energy delivered to the capsule may be insufficient to trigger fusion ignition. Mid-Z Ion-driven fast ignition generates an ion beam that delivers extra energy to the pre-compressed capsule to start the fusion process.



Achieving mid-Z ion fast ignition places stringent requirements on ion beams

- High energy
- Conversion efficiency

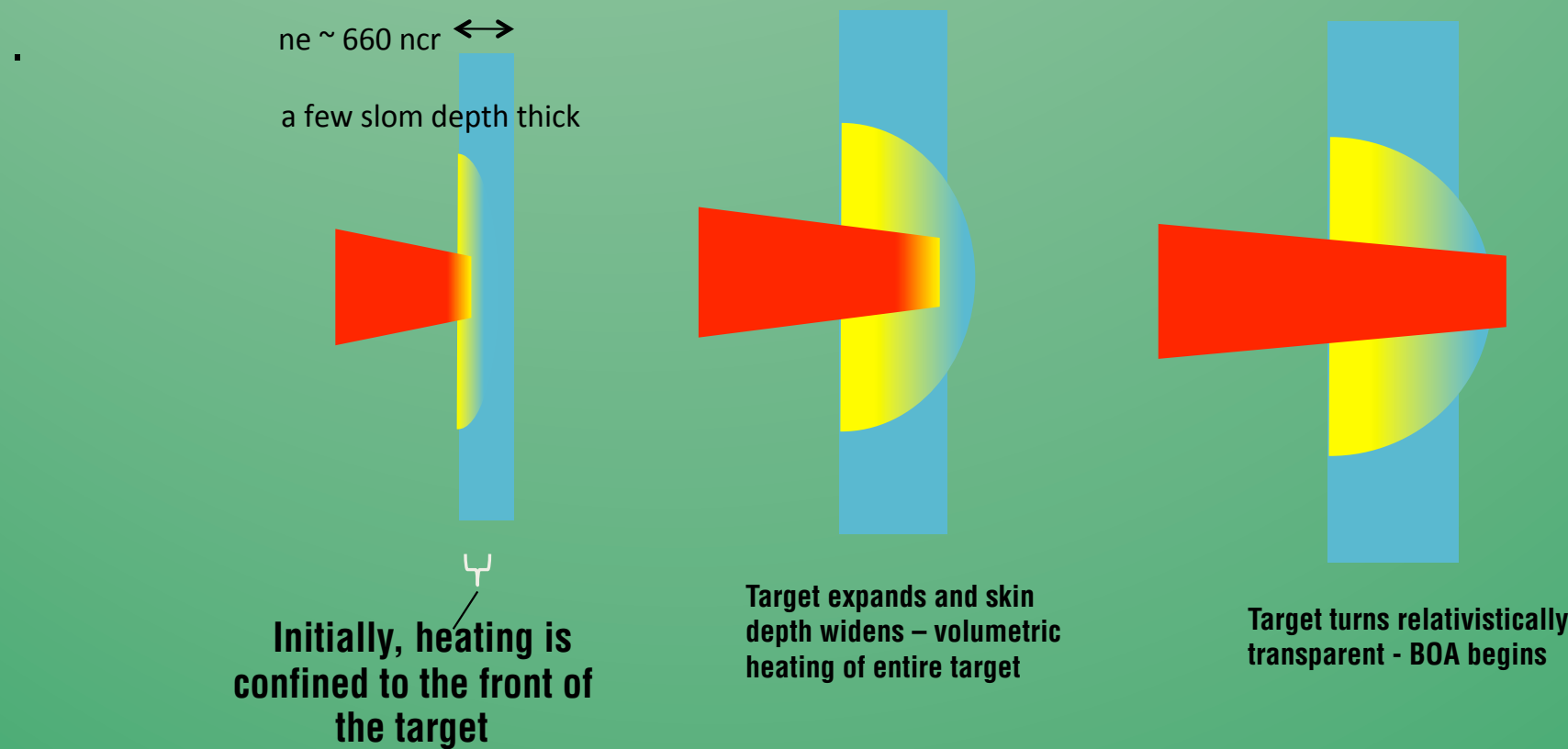
The Break-Out Afterburner (BOA) is one acceleration technique that may achieve these desiderata

- Acceleration of carbon ions to greater than 2 GeV energy
- Requires much lower laser intensities which have already been realized experimentally (10^{21} W/cm²)

Our research on the BOA is enabled by recent advances in plasma simulation and experimental capability

- High performance computing – enables 3D modeling of BOA physics
 - Highly optimized 3D PIC code VPIC
 - Powerful supercomputers (e.g., Roadrunner)
- Two new experimental technologies enabling realization of BOA in the laboratory
 - High contrast, high energy pulses at Trident
 - Free-standing nm-targets at LMU

A new acceleration regime emerges: The break-out afterburner (BOA)



3D VPIC simulation of the BOA done on Roadrunner

